PICK Quick Reference

DSS-10482-00, Rev. A00

PICK
Quick Reference

This booklet is your quick reminder guide for Pick64+ version 2.3 and later. It contains brief descriptions of the PICK commands and features you use every day, including PICK/BASIC, JET, PROCs, ACCESS verbs, and more.

This booklet is not intended to teach you how to use any aspect of PICK. Rather, it is for people who are already somewhat familiar with PICK concepts, and need an occasional reminder about command syntax, ACCESS structure, and the like. For more detailed information about any of the features mentioned here, refer to these other parts of the Pick64+ documentation library:

- *PICK User's Guide*
- *PICK/BASIC Reference Manual*
- *PICK System Operator's Guide*
- *Jet Word Processor User's Guide*
- *PICK Library Master Index*

TABLE OF CONTENTS

ACCESS Sentence Structure 1
ACCESS Verb List 3
List Processing Verbs 3
Editor Commands 4
JET Word Processor 6
PROC Commands 9
Spooler Verbs 10
Tape Handling Verbs 13
TANDEM and MIRROR Verbs 15
PICK/BASIC TCL Verbs 16
PICK/BASIC Statements and Functions 16
PICK/BASIC Debugger Commands 24

© 1993 Alpha Microsystems

REVISIONS INCORPORATED	
REVISION	**DATE**
00	28 October 1993

PICK Quick Reference
To re-order this document, order part number DSS-10482-00

This document applies to Pick64+ 2.3 (G00) and later.

The information contained in this manual is believed to be accurate and reliable. However, no responsibility for the accuracy, completeness, or use of this information is assumed by Alpha Microsystems.

This document was written by Robert Shelton.

PICK is a registered trademark of PICK Systems.
ACCESS is a trademark of PICK Systems.

ALPHA MICROSYSTEMS
3511 Sunflower
P.O. Box 25059
Santa Ana, CA 92799

ACCESS SENTENCE STRUCTURE

The general form of an ACCESS sentence is:

```
verb file {items} {selection-crit} {sort-keys} {output specs}
{print limiters} {modifiers} {(options}
```

Very few ACCESS verbs actually use all parts of the sentence structure.

file can be: filename
DICT filename
DATA filename
dictname,dataname

items can be a list of item-IDs enclosed in single or double quotes or backslashes, and/or use these operators:

= or EQ	# or NE	< or LT	> or GT
<= or LE	>= or GE	AND	OR

Items can also be determined by an active select-list.

selection-crit can check for the existence of an attribute and/or use the operators above. It must start with WITH and can also use:

EVERY or EACH	All values must pass test (with NO, all must fail).
NO	Value must fail test.

sort-keys can be:

BY att	Ascending sort by ***att.***
BY-DSND att	Descending sort by ***att.***
BY-EXP att	Ascending exploding sort (by values within ***att***).
BY-EXP-DSND att	Descending exploding sort (by values within ***att***).

output spec defines the attributes to print.

print limiters selects values from multi-valued attributes. The format is similar to selection-crit but does not start with WITH.

modifiers are general modifiers. They may be:

BREAK-ON att {"text"} Subtotal on specified ***att.*** ***text*** prints next to break totals. Text may include following options in single quotes:

B	Use attribute with HEADING/FOOTING B option.
D	Do not break after single line.
L	Suppress blank line before break data.
P	Page after break.
R	Inhibit page rollover.
U	Underline totals.
V	Insert control-break value here.

ACCESS SENTENCE STRUCTURE, continued

COL-HDR-SUPP	Don't print column headings.
DBL-SPC	Double space.
DET-SUPP	Print totals only.
FOOTING "text"	Print ***text*** as footer. May include following in single quotes: B Insert control-break value. C Center line. D Print date. F Print file name. Fn Print file name left justified in ***n*** blanks. L Start new line. P Print page number right justified. Pn Print page number left justified in ***n*** blanks. T Print time and date.
GRAND-TOTAL "text"	Print ***text*** on grand total line.
HEADING "text"	Print ***text*** as page header. Can use same special characters as FOOTING.
HDR-SUPP	Don't print time/date and page number.
ID-SUPP	Don't list item-IDs.
LINE-SUPP	Don't print line numbers.
LPTR	Print to spooler.
ONLY	Don't display default attributes.
NI-SUPP	Don't print number of items.
NOPAGE	Don't stop after each page.
TAPE	Read file from T-DUMP tape.
TOTAL att {"limit"}	Print total of ***att***. ***limit*** is the same as print limiters, above.

Most of the ***options*** provide alternate ways to perform the same functions as the modifiers:

- C Same as COL-HDR-SUPP.
- D Same as DET-SUPP.
- F Start each item on new page (COPY, LIST-ITEM and SORT-ITEM only).
- H Same as HDR-SUPP.
- I Same as ID-SUPP.
- J Same as NI-SUPP
- N Same as NOPAGE.
- P Same as LPTR.
- S Same as LINE-SUPP.
- S When used with select-list, suppresses `Not on file` message.

ACCESS VERB LIST

COUNT	Counts items.
HASH-TEST	Determines modulo for a file.
ISTAT	Prints file use information and histogram.
LIST	Lists selected items and attributes.
LIST-ITEM	Prints items to terminal or spooler, like COPY verb.
LIST-LABEL†	Prints labels for items.
QSELECT	Creates select-list from attributes or values.
REFORMAT	Writes selected items and attributes to file or tape.
S-DUMP*	Same as T-DUMP, but sorts items before copying.
SELECT	Creates select-list.
SORT	Same as LIST, but sorts items before listing.
SORT-ITEM	Same as LIST-ITEM, with sort.
SORT-LABEL†	Same as LIST-LABEL, but sorts before printing.
SREFORMAT	Same as REFORMAT, with sort.
SSELECT	Creates sorted select-list.
STAT	Same as SUM, but also gives number of items and average value.
SUM	Totals attribute values.
T-DUMP*	Copies selected items to tape.
T-LOAD*	Retrieves items from T-DUMP tape.

† First prompt asks for (in this order): labels per row, rows per label, lines between labels, left indent, width of label, columns between labels, C to skip null attributes.

* To use item-IDs with these verbs, you must enclose them in quotes.

LIST PROCESSING VERBS

Three of the ACCESS verbs listed above, SELECT, QSELECT, and SSELECT, create select-lists for use by other processors. The following verbs also deal with lists (***items*** in these descriptions indicates one or more item-IDs):

`CL{IST} file items`	Copies ***items*** in ***file*** as list items.
`COPY-LIST items`	Copies ***items*** in POINTER-FILE as list items.
`DELETE-LIST items`	Deletes ***items*** from POINTER-FILE. Use on lists only.
`DL{IST} file items`	Deletes ***items*** from ***file***. Works only on list items.
`EDIT-LIST items`	Edits ***items*** in POINTER-FILE and saves as list items.
`EL{IST} file items`	Edits ***items*** in ***file*** and saves as list items.
`GET-LIST items`	Retrieves ***items*** from POINTER-FILE for use.
`GL{IST} file items`	Retrieves ***items*** from ***file*** for use.
`SAVE-LIST item`	Saves current select-list as ***item*** in POINTER-FILE.
`SL{IST} file item`	Saves current select-list as ***item*** in ***file***.

You can use all of these verbs except DELETE-LIST and DLIST on either list items or regular items; the result is always a list item. Similarly, using a regular TCL verb, such as COPY or EDIT, on a list item results in a regular item, not a list.

For a discussion of list item format and Pick64$^+$'s R83- and OA-compatible list formats, see the *PICK User's Guide*.

EDITOR COMMANDS

`A`
Repeats last locate command.

`AL(n)/string/`
Appends ***string*** to the next ***n*** lines.

`AS`
Toggles assembly format option.

`B`
Goes to end of item.

`BL(n)/string/`
Breaks next ***n*** lines at the end of ***string***.

`C`
Displays column ruler.

`DE(n)`
Deletes ***n*** lines, starting with current line.

`DE(n)"string"(p(-q))`
Searches columns ***p-q*** of next ***n*** lines for ***string*** and deletes all lines containing it.

`EX(K)`
Stops editing without saving item. ***K*** cancels select-list and leaves editor.

`F`
Flips buffers.

`FD(K)`
Deletes item from file. ***K*** is as in EX command, above.

`FI(K)`
Writes item back to current file and stops editing. ***K*** is as in EX command, above.

`FI(K)(O) (file) item`
Files item as ***item*** in ***file***. ***O*** overwrites existing item. ***K*** is as in EX command, above.

`FS`
Files item and continues editing.

`FS(O) (file) item`
Files item as ***item*** in ***file*** and continues editing. ***O*** overwrites existing item.

`(G)n`
Goes to line ***n***.

`HX`
Toggles hexadecimal output. Does not work on multivalued (assembly language) items.

`I`
Enters input mode.

`I text`
Inserts ***text*** as new line after current line.

`JL`
Joins current line to previous.

`L(n)`
Lists ***n*** lines after current line.

`L(n)"str" (p(-q))`
Looks for ***str*** in columns ***p-q*** of next ***n*** lines. Without n, finds next occurrence of str.

EDITOR COMMANDS, continued

`LC`
Toggles display of object code.

`M`
Toggles macro expansion.

`ME{n}/item/{m}`
Inserts ***n*** lines from ***item***, starting with line ***m***, after current line.

`ME{n}(file {item}) {m}`
Same as above, but ***item*** is in ***file***.

`N{n}`
Increments line counter by ***n*** and lists new line.

`Pn command`
Stores ***command*** as prestore command ***n***.

`Pn`
Executes prestored command ***n***.

`PD`
Displays prestored commands.

`R{n}`
Replaces ***n*** lines starting with current line.

`R{n}/str1/str2/ {p{-q}}`
Searches columns ***p-q*** of next ***n*** lines; replaces first occurrence of ***str1*** in each with ***str2***.

`RU{n}/str1/str2/ {p{-q}}`
Same as above, but replaces all occurrences.

`S`
Toggles line number display.

`S?`
Displays size of the item.

`TB n{,n...}`
Sets tab stop(s) at column(s) ***n***.

`T`
Goes to beginning of the item.

`U{n}`
Moves up ***n*** lines.

`X{F}`
Cancels last input, insert, delete or replace. With ***F***, cancels all changes since buffers were flipped.

`Z{p{-q}}`
Causes L command to show only columns ***p-q***.

`?`
Displays the file name, item, and current line number.

`^`
Toggles wildcard character.

JET WORD PROCESSOR

JET Commands

To edit or create a word processing document: `JIN file {items} {(V}`
To edit or create a program: `JED file {items} {(V}`

Both these verbs open the indicated ***items*** in ***file*** for editing. If you include more than one item, each is opened consecutively when you leave the previous one. If you don't include an item list, items are taken from an active select-list. The ***V*** option starts in view off mode.

To display or print a document: `JOUT file {items} {(opts}`

JOUT displays the listed ***items*** from ***file*** consecutively. ***opts*** can be:

- C Output to console or slave printer.
- E Print to NEC Spinwriter. Must also use Q option.
- F Form feed before each item.
- L Print line numbers.
- N Don't stop after each page.
- P Print to spooler.
- Q Print to letter quality printer.
- R Don't use random blanks for right justification.
- S Proportional spacing. Must also use Q option.
- U Print to QUME Sprint printer. Must also use Q option.
- V Print rulers, command lines and control characters.

Editing Text

KEY	FUNCTION	KEY	FUNCTION
A	Another search/replace	Q	Delete one word
C	Redisplay screen	R	Replace characters
D	Delete character	S	Search and replace
E	Edit or add ruler	U	Spell check at input
FD	Delete item	V	View screen
FI	Save item and exit	W	Replace word
FX	Exit item	X	Exit document
FS	Save item	Y	Delete sentence
FK	Exit entire file	Z	Transpose characters
G(n)	Go to line ***n***	[]	Cut text
I	Insert text	"	Display cut text
K	Delete one line	:	Change case of letter
L	Insert new line	*	Change case of word
M	Merge	|	Change case of line
N	Display line numbers	!	Check spelling
O	Display output	ESC	Return to edit mode
P	Paste	?	Display editing help

JET WORD PROCESSOR, continued

Moving Through Your Document

KEY	FUNCTION	KEY	FUNCTION
0	Back one page	9	Forward one word
1	Back one sentence	B	Bottom of document
2	Forward one line	T	Top of document
3	Forward one sentence	ENTER	Beginning of line
4	Back one character	^	End of line
5	Top of current page	.	Forward one page
6	Forward one character	}	Forward one paragraph
7	Back one word	{	Back one paragraph
8	Back one line		

Changing a Ruler

Use SPACE and BACKSPACE to move right and left on the ruler. Use these keys to edit the ruler:

KEY	FUNCTION	KEY	FUNCTION
L	Set left margin	<	Set left tab
>	Set right tab	.	Set decimal tab
A	Set automatic indent tab	C	Set center tab
Z	Set hyphenation zone	R	Set right margin
B	Move left margin to here; move right margin equal distance in opposite direction.	S	Save ruler. Prompts for number (0 - 9).
G	Get ruler. Prompts for number (0 - 9).	-	(dash) Erase character
_	(underline) Clear ruler	X	Exit ruler without embedding it
?	Display ruler help	ESC	Embed ruler and return to editing mode
ENTER	Go to far left of ruler	TAB	Move to next ruler character to right

Boldface and Underlining

While entering text, use these characters:

CTRL-B	Start boldface.
CTRL-D	End boldface.
CTRL-V	Underline, including blanks.
CTRL-W	Underline characters only.
CTRL-X	End underline.

JET WORD PROCESSOR, continued

Document Formatting

All formatting commands must be alone on a line and start with a backslash. This table gives only the shortest form of each formatting command:

`\BP {n}`	Begins new page; sets page number to ***n***.
`\CH {file} id`	Begins printing item ***id*** from ***file*** or from current file. Does not resume current document.
`\F`	Following text (until next command) is footing. Footing text may include these characters in single quotes: D Date I Item-ID P Page number T Time and date
`\HILITE c p`	Prints ***c*** two spaces outside margin ***p***. p can be L for left, R for right or B for both. Continues until \HILITE OFF.
`\HILITE OFF`	Ends highlighting.
`\INDENT n`	Indents ***n*** characters.
`\INDEX.FILE file`	Specifies ***file*** for the \INDEX command.
`\INDEX text`	Creates index entry of ***text***.
`\J`	Begins right justification.
`\NJ`	Turns off right justification.
`\NUMBER n`	Sets page number to ***n***.
`\PAGE n`	Same as \NUMBER.
`\PAGE LENGTH n`	Sets page length to ***n***.
`\PAGE WIDTH n`	Sets page width to ***n***.
`\PA`	Pauses until key is pressed.
`\PC`	Closes printer; forces spooler to output.
`\PF n`	Prints to print file ***n***.
`\PI n`	Sets pitch to ***n*** characters per inch.
`\PR text`	Uses ***text*** as prompt for next ~Input or \PA command.
`\R file id`	Prints item ***id*** from ***file***, then resumes this document.
`\SP n`	Sets line spacing to ***n***.
`\TESTPAGE n`	Page break if less than ***n*** lines are left.
`\TITLE.FILE file`	Uses ***file*** for \TITLE commands.
`\TITLE`	Next line is entry for title file.
`\XREF.FILE file`	Uses ***file*** for cross-reference file.
`\XREF text`	Creates cross-reference entry of ***text***.
`\* text`	***text*** is comment.

Inserting Special Text

`~INPUT`	Pauses for you to enter text from the keyboard.
`~INSERT file att`	Inserts the specified ***att*** from the next item in an active select-list created from ***file***.
`~READNEXT`	Inserts next value from active select-list.
`~TIME`	Inserts current time.

PROC COMMANDS

If not stated otherwise, these commands affect the currently active input and/or output buffer. Only the shortest form of each command is given.

`A{c}{p}{,n}`
Moves the first ***n*** characters of parameter ***p*** from input to output buffer. ***c*** is the surround character for the primary output buffer.

`A{c}(n,m)`
Starting with character ***m***, moves ***n*** characters to output buffer. ***c*** is the surround character, as above.

`B`
Moves the input pointer back one parameter.

`BO`
Moves the output pointer back one parameter.

`C or *`
Comment.

`D {p{,n}}{+}`
Displays the first ***n*** characters of parameter ***p*** on terminal. **+** suppresses carriage return at end.

`F`
Moves the input pointer forward one parameter.

`G {n or A}`
Transfers control to the command with label number ***n*** or, with ***A***, to the command with the label equal to the parameter pointed to in the input buffer.

`H text{<}`
Places ***text*** in the output buffer. **<** is a carriage return.

`IF`
Conditional command execution. Can check input parameters, pattern match, error numbers, or a select-list.

`IH text`
Replaces the current input parameter with text.

`IP{r}`
Inputs data from the terminal. ***r*** is optional prompt character.

`IS{r}`
Selects and clears the secondary input buffer and inputs data from the terminal. ***r*** is an optional prompt character.

`IT`
Clears the active input buffer and inputs tape label and reel number.

`Otext{+}`
Displays ***text*** on the terminal. **+** suppresses ending carriage return.

`P {option}`
Executes TCL command from the primary output buffer; stack is input. Options are:

- H Suppress terminal output.
- P Display both output buffers.
- W Display buffers, then request confirmation.
- X Exit PROC after executing command.

PROC COMMANDS, continued

`RI{p}`
Clears the entire secondary input buffer and the primary input buffer beginning at parameter ***p***. Without p, clears whole primary buffer.

`RO`
Clears both output buffers and selects the primary output buffer.

`RTN`
Returns to the calling PROC from a subroutine.

`Sn`
Sets the input pointer to parameter ***n***.

`SP`
Selects the primary input buffer.

`SS`
Selects the secondary input buffer.

`STON`
Selects the secondary output buffer (stack).

`STOFF`
Selects the primary output buffer.

`T`
Outputs formatted text. Many functions are available.

`Uexit`
Executes user exit ***exit***.

`X{text{+}}`
Ends PROC. Displays ***text***, if any. + at end of text suppresses carriage return.

`+n`
Adds ***n*** to the current input parameter.

`-n`
Subtracts ***n*** from the current input parameter.

`(file {item}) {n}`
Links to the PROC named ***item*** in ***file***. Starts executing with line labeled ***n***. If you omit item, the current input parameter is the item-ID.

`[{file {item}}] {n}`
Calls the named PROC as a subroutine. Without ***file*** and ***item***, calls a subroutine in this PROC starting with the line labeled ***n***.

SPOOLER VERBS

SP-STATUS: Displays information on the spooler and selected printers. Options are:

`none`	Include all allocated printers.
`B`	Include unallocated printers.
`n{-m}`	Include printers ***n-m*** only.
`P`	Print output.

SPOOLER VERBS, continued

LISTABS: Lists spooler assignments. Options are:

`n{-m}`	Display line ***n*** or lines ***n-m***.
`N`	Don't pause at end of each screen.
`P`	Print output.

LISTPEQS: Displays print file information. Options are:

`n{-m}`	Display file ***n*** or files ***n-m***.
`'acc'`	Display files created on account ***acc***.
`A`	Display files generated by current account.
`C`	Display totals only.
`E`	Display print file storage locations.
`F`	Display files sequentially by form queue.
`L`	Include deleted print files.
`P`	Print output.

LISTPTR: Displays printer information. Options are:

`n{-m}`	Display printer ***n*** or printers ***n-m***.
`B`	Include unallocated printers.
`N`	Don't pause at end of each screen.
`P`	Print output.

STARTPTR: Defines printer. Format is: `STARTPTR n,{(}forms{)},p,tm,{A}{(XyS)}`

`n`	Printer number.
`forms`	Up to 3 form queue numbers, separated by commas. If more than one, must be enclosed in parentheses.
`p`	Pages to eject after file.
`t`	Type P for parallel or S for serial.
`m`	For serial printer, line number; for parallel, device number.
`A`	Align forms.
`X`	Serial printer doesn't recognize page eject.
`y`	Lines per page; use only with X.
`S`	Ignore initial form feed for serial printer.

Note: To restart a printer after STOPPTR, you need to include only the printer number.

SP-ASSIGN: Changes spooler assignment for this line. Options are:

`n`	Print ***n*** copies.
`C`	Choke output. Works only with I.
`Fn`	Print to form queue ***n***.
`H`	Create hold file.
`I`	Begin printer output immediately.
`M`	Suppress entry number message.
`O`	Leave print file open when done.
`Rn`	Open file ***n*** for use by BASIC.
`S`	Do not send output to printer.
`T`	Send print file to tape.
`?`	Display spooler assignment.

SPOOLER VERBS, continued

SP-EDIT: Edits hold files. Selection options are:

`n{-m}`	Select hold file ***n*** or files ***n-m***.
`Fn{-m}`	Select files for form queue ***n*** or queues ***n-m***.
`U`	Select all files.
`'acc'`	Select all files from account ***acc***.

Output options are:

`none`	Send file to queue assigned when printed or to tape if current assignment is tape.
`R`	Send each file to currently assigned queue.
`P`	Send file to queue assigned when printed.
`T`	Send file to tape.
`W`	Send to tape if currently assigned.
`H`	Send to tape without tape label (may combine with T or W).
`TW`	Send to tape; wait for tape if unavailable.
`L`	Look at first screen of file.
`LO`	Look at first screen of file, even if being printed.
`MS`	Send to selected output without prompting.
`MD`	Delete selected files without prompting.
`V`	Don't delete trailing blanks from data file.
`N`	Don't pause after each screen on display.

Note: Unless you use MS or MD, you are prompted for the disposition of each file.

SP-KILL: Stops print files during printing, removes files from form queue, or deletes printer from spooler. Action options are:

`none`	End current print job.
`F`	Remove print file(s) from queue.
`D`	Delete printer(s) from spooler.
`N`	Do not print abort message.
`O`	With F only. Remove files from queue during output.

Selection options refer to printers or, with the F option, print files. They are:

`n{-m}`	Number ***n*** or ***n-m***.
`B`	All possible.
`A`	All printers printing files created by this account.

STOPPTR: Stops printer after the current job. Options are:

`n{-m}`	Stop printer ***n*** or printers ***n-m***.
`B`	Stop all printers.
`D`	Delete printer (as in SP-KILL). Comes before printer number.
`W`	Wait until printer is inactive.

TAPE HANDLING VERBS

To select and attach a tape device:

SET-DAT	DAT drive (Procom).
SET-AHADAT	DAT drive (Adaptec).
SET-1GB	Tandberg 4100 (Procom).
SET-AHA1GB	Tandberg 4100 (Adaptec).
SET-TDC	Tandberg 3800 (Procom).
SET-AHATDC	Tandberg 3800 (Adaptec).
SET-AHATD	Tandberg 3600 (Adaptec).
SET-SCT	Wangtek 1/4" tape.
SET-TCC	Exabyte 8mm.
SET-HALF	1/2" mag tape.
SET-FLOPPY	Diskette. Options: A or B: Drive name. H: High density S: Standard density
T-ATT {x}{U}	Attaches currently set tape device and sets block size to ***x***. ***U*** attaches even if tape is attached to other line.
T-DET {U}	Detaches tape device. ***U*** detaches it from any line.

Tape Utilities:

T-BCK {x}	Backspaces to previous EOF or ***x*** blocks.
T-FWD {x}	Moves forward to next EOF or ***x*** blocks.
T-FSF	Moves to just past next EOF.
T-FSR {x}	Moves forward one block or ***x*** blocks.
T-BSF	Moves back to just before previous EOF.
T-BSR {x}	Moves back one block or ***x*** blocks.
T-SPACE x	Moves forward ***x*** files.
T-EOD	Moves to end of data.
T-REW	Rewinds the tape.
T-WEOF	Writes an EOF mark.
T-CHK {{A}	Checks current file for parity errors; with ***A***, checks to EOD.
T-UNLD	Unloads 1/2" tape; rewinds others.
T-RETEN	Retensions 1/4" tape.
T-ERASE	Erases and retensions 1/4" tape.
T-RESET	Resets SCSI tape device. Works on Procom controller only.
T-FORMAT	Formats a diskette.
T-STATUS	Displays type and block size of currently assigned tape device.

NOTE: many of these verbs work only on tape devices which support backspace commands.

TAPE HANDLING VERBS, continued

Tape Input/Output Verbs:

`T-DUMP file {'items'} {sel-crit} {HEADER "name"} {(opts}`
`S-DUMP file {'items'} {sel-crit} {keys} {HEADER "name"}{(opts}`

Writes selected items from file to tape; S-DUMP sorts the items before writing. Parts of sentence are as for other ACCESS verbs (but Item-IDs must be in quotes). HEADER includes ***name*** in tape label. Options are:

- B for backward (pre-2.3) compatible tape.
- C for R83 compatible tape.
- I not to display item-IDs.

`T-LOAD file {'items'} {sel-crit}} {(opts}`

Writes selected items from tape to file. Parts of sentence are as for other ACCESS verbs (but Item-IDs must be in quotes). Options are:

- I not to display item-IDs.
- O to overwrite existing items.

`T-READ {n{-m}} {(options}`

Displays or prints current file from tape or blocks ***n-m*** from current tape position. Options are:

- A to convert EBCDIC to ASCII.
- P to print.
- X to display in hexadecimal.

`T-RDLBL`

Displays tape label from current tape position. Has no options.

`T-WTLBL`

Writes tape label at current tape position. Prompts for label text.

`SP-TAPEOUT {(options}`

Moves print file from tape to spooler. Options are:

- A to convert EBCDIC to ASCII.
- L to treat each block as a single line.
- U to make letters upper case.

TANDEM AND MIRROR VERBS

`MIRROR {port#{,port#,...}}{(options}`

Attaches one or more guests, specified by ***port#***, to a host terminal; the terminal issuing the MIRROR becomes the host process. Default is interactive mode. Options are:

- C to prevent character loss due to higher throughput of host terminal.
- L to turn off interactive mode; discards all guest input except exit hotkey.
- S to suppress exit hotkey. Connection can only be broken by TANDEM-ABORT or TANDEM-CLEAR.

`TANDEM {port#}{(options}`

Attaches this terminal as a guest to ***port#,*** or performs action of the entered ***option***. Default is interactive mode. Options are:

- ? to display all tandem/mirror relationships.
- B to block this port from being a host or guest.
- C to prevent character loss due to higher throughput of host terminal.
- E to define the exit hotkey.
- L to turn off interactive mode; discards all guest input except exit hotkey.
- S to suppress exit hotkey. Can only break connection with TANDEM-ABORT or TANDEM-CLEAR.
- U to enable tandem for specified port after it has been blocked.
- X to detach specified guest terminal from its port.
- Z to display the tandem/mirror status of all ports.

Using TANDEM with no port number is the same as the ? option.

`TANDEM-ABORT {port#}`

Terminates a tandem/mirror connection for the specified ***port#***. If you don't include a port number, it affects the current port.

`TANDEM-CLEAR`

Breaks all tandem/mirror connections on the system. It does not accept any arguments.

`TANDEM-DISABLE {port#}`

Blocks the specified ***port#*** from all tandem/mirror connections, as host or guest. If you don't include a port number, it affects the current port.

`TANDEM-ENABLE {port#}`

Unblocks the specified ***port#;*** allows all tandem/mirror connections, as host or guest. If you don't include a port number, it affects the current port.

`TANDEM-ESCAPE {port#}`

Same as TANDEM-ABORT.

`TANDEM-ESC-KEY {port#}`

Sets the exit hotkey for ***port#***. A prompt appears for you to enter the single character (usually a control character) you want. If you don't include a port number, it affects the current port.

PICK/BASIC TCL VERBS

All these commands take a file name and program (item) name:

BASIC	Compiles source code.
COMPILE	Compiles source code.
CATALOG	Catalogs object code into MD.
DECATALOG	Decatalogs program.
RUN	Runs program.

PICK/BASIC STATEMENTS AND FUNCTIONS

The following list gives the format and meaning of all PICK/BASIC statements and functions, in alphabetical order. It uses these abbreviations:

fvar indicates a file variable assigned to an open file. If fvar is optional and omitted, the file most recently opened without a file variable is used.

a indicates an attribute, ***v*** a value, and ***s*** a subvalue.

da stands for a dynamic array.

@

`@(col,row)`
Moves the cursor to the indicated position.

`@(-n)`
Invokes special display attribute ***n***.

A

`ABORT {en {,p1,p2...}}`
Ends the program, including any calling PROC. Displays error message number ***en*** using parameters ***p1***, etc.

`ABS(exp)`
Absolute value.

`ALPHA(exp)`
Returns true if ***exp*** is an alphabetic string, false otherwise.

`ASCII(exp)`
Converts ***exp*** from EBCDIC to ASCII.

B

`BREAK ON`
Turns on the debugger.

`BREAK OFF`
Turns off the debugger.

`BREAK EXP`
Turns the debugger off if ***exp*** = 0, on otherwise.

PICK/BASIC STATEMENTS AND FUNCTIONS, continued

C

`CALL name {args}`
Calls the external subroutine ***name***, passing ***args*** to it.

`CASE`
Conditional branching. Branches to statements following the true ***CASE exp*** statement. Format is:

```
BEGIN CASE
CASE exp
statements
...
END CASE
```

`CHAIN "string"`
Executes ***string*** as a TCL command.

`CHAR(num)`
Changes ***num*** to its ASCII equivalent.

`CLEARFILE {fvar}`
Clears data from the file opened as ***fvar***.

`COL1()`
Returns column before the last FIELD substring.

`COL2()`
Returns column after the last FIELD substring.

`COS(exp)`
Cosine.

`COUNT(str1,str2)`
Counts occurrences of ***str1*** in ***str2***.

`CRT list`
Prints values of expressions in ***list*** on the screen. May include cursor control (`@`) functions.

D

`DATA exp,exp,...`
Stacks data in ***exp***s for use with CHAIN or EXECUTE.

`DATE()`
Returns the internal date as a string.

`DCOUNT(str,del)`
Counts the number of fields separated by ***del*** in ***str***.

`DEBUG`
Calls the BASIC debugger.

`DEL da <a{,v{,s}}>` or `DELETE (da,a{,v{,s}})`
Deletes the specified part of ***da***.

`DELETE{fvar,} item`
Deletes ***item*** from the file opened as ***fvar***.

`DISPLAY list {:}`
Synonym for CRT.

PICK/BASIC STATEMENTS AND FUNCTIONS, continued

`DTX(exp)`
Converts ***exp*** from decimal to hexadecimal.

E

`EBCDIC(exp)`
Converts ***exp*** from ASCII to EBCDIC.

`ECHO {ON} {OFF} {exp}`
Turns terminal echoing on or off. With ***exp***, turns off if exp = 0, on otherwise.

`END`
Physical end of program.

`ENTER prog`
Transfers control to ***prog***. prog may be a literal in quotes or a variable preceded by @.

`EXECUTE exp {CAPTURING v1} {RETURNING v2}`
Executes ***exp*** as a TCL statement. Output from the statement is returned in ***v1***, error numbers in ***v2***.

`EXIT`
Exits a conditional construct such as LOOP or CASE.

`EXP(power)`
Raise *e* to ***power***.

`EXTRACT (da,a{,v{,s}}) or da<a{,v{,s}}>`
Extracts the specified part of ***da***.

F

`FIELD(exp,del,occ)`
Returns a substring delimited by ***del*** from ***exp***. ***occ*** specifies which occurrence of del ends the selected substring.

`FMT(exp1,exp2)`
Formats ***exp1*** using the format in ***exp2***.

`FOOTING "text'opts'"`
Prints ***text*** on the bottom of each page. ***opts*** may be anywhere within the text. Available options are:

- C Center line.
- D Print date.
- L Carriage return/Line feed.
- P Print page number.
- T Print time and date.

`FOR var = exp to exp {STEP exp} {WHILE or UNTIL exp}`
Starts a FOR...NEXT loop. Lasts until ***var*** equals second ***exp*** or ***WHILE*** or ***UNTIL*** condition is met. An EXIT statement leaves the loop.

G

`GO{TO} label`
Transfers control to ***label***.

`GOSUB label`
Calls the subroutine starting with ***label***.

PICK/BASIC STATEMENTS AND FUNCTIONS, continued

H

`HEADING "text'opts'"`
Prints ***text*** at the top of each page. ***opts*** are the same as for FOOTING.

I

`ICONV(exp,con)`
Applies input conversion ***con*** to ***exp***. Cannot use ACCESS F conversions.

`IF exp {THEN stmts} {ELSE stmts}`
Evaluates ***exp***; performs ***stmts*** after THEN if true, after ELSE if not.

`IN var`
Accepts one character from keyboard into ***var***.

`INCLUDE {file} item`
Compiles ***item*** as part of this program.

`INDEX(str,sub,occ)`
Returns the starting column of ***sub*** within ***str***. ***occ*** specifies which occurrence of sub you want.

`INPUT var{,int{_}} {:} {THEN/ELSE stmts}`
Accepts keyboard input into ***var***. ***int*** is a maximum number of characters. Underline suppresses auto-carriage return after int characters. Colon suppresses carriage return/line feed after input.

`INPUT @(x,y):var {mask} {THEN/ELSE stmts}`
Positions cursor at ***x,y*** and accepts input into ***var***. Input is verified against the format ***mask***.

`INPUTERR exp`
Prints ***exp*** as error message.

`INPUTTRAP 'x1x2x3...' GOSUB n1,n2,n3`
When ***x1*** is entered, goes to subroutine with label ***n1, etc.***

`INPUTTRAP 'x1x2x3...' GOTO n1,n2,n3`
When ***x1*** is entered, goes to label ***n1, etc.***

`INPUTNULL x`
When ***x*** is entered, sets default to null string.

`INSERT(da,a{,v{,s});exp)`
Inserts ***exp*** into ***da*** as an attribute, value, or subvalue.

`INT(exp)`
Truncates the value of ***exp*** to an integer.

K

`KEY(op,root,key,id) THEN/ELSE statements`
Performs ***op*** on item with the secondary key ***key***. ***root*** is root FID of the secondary index. ***id*** holds returned item-ID. op may be:

- C Find closest match.
- N Read next key.
- P Read previous key.
- R Return exact match only.

PICK/BASIC STATEMENTS AND FUNCTIONS, continued

L

`LEN(exp)`
Number of characters in ***exp***.

`LN(exp)`
Natural log.

`LOCK exp {THEN/ELSE statements}`
Sets execution lock number ***exp***. ELSE statements are performed if the lock is already set.

`LOOP stmts {WHILE or UNTIL exp} {stmts} REPEAT`
Program loop. An EXIT statement leaves the loop.

M

`MATREAD{U} array FROM{fvar,}item LOCKED stmts THEN/ELSE stmts`
Assigns each attribute of ***item*** in ***fvar*** to successive elements in ***array***. ***U*** leaves item locked.

`MATWRITE{U} array ON {fvar,} item`
Writes elements of ***array*** into attributes of ***item*** in ***fvar***. ***U*** leaves item locked.

`MOD(num,denom)`
Divides ***num*** by ***denom*** and returns the remainder.

N

`NEXT`
Returns to beginning of FOR...NEXT loop.

`NOT(exp)`
Returns the logical inverse of ***exp***.

`NULL`
No operation. Can be used whenever a statement is required.

`NUM(exp)`
Returns true if ***exp*** is a number, false otherwise.

O

`OCONV(exp,con)`
Applies output conversion ***con*** to ***exp***. Cannot use ACCESS F conversions.

`ON exp GOSUB lab1, lab2, ...`
Calls subroutine ***lab1, lab2*** etc., based on the value of ***exp***.

`ON exp GOTO lab1, lab2, ...`
Branches to ***lab1, lab2*** etc., based on the value of ***exp***.

`OPEN file {TO fvar} THEN/ELSE statements`
Opens file ***file*** and assigns it to variable ***fvar***. If no fvar is entered, a default variable is assigned.

`OUT var`
Displays on the terminal the character with the ASCII value ***var***.

PICK/BASIC STATEMENTS AND FUNCTIONS, continued

P

`PAGE {exp}`
Prints the footing, causes a page break, and prints the heading on the next page. ***exp*** becomes the next page number.

`PRECISION{num}`
Sets numeric precision for the program. ***num*** can be from 0-6; default is 4.

`PRINT {ON exp} list`
Prints the value in ***list*** to the current output device (as set by PRINTER statement). ***ON exp*** prints to print file number exp.

`PRINTER CLOSE`
Starts spooler output of open print file.

`PRINTER ON`
Sets the output device to the printer.

`PRINTER OFF`
Sets the output device to the terminal.

`PROMPT exp`
Sets the input prompt to ***exp***.

`PWR(exp,power)`
Raises **exp** to **power**.

R

`READ{U} var FROM {fvar,} item LOCKED stmts THEN/ELSE stmts`
Reads ***item*** from ***fvar*** into ***var***. ***U*** locks item.

`READNEXT var {,vmc} {FROM selvar} THEN/ELSE statements`
Reads next item-ID from the select-list assigned to ***selvar*** into ***var***. ***vmc*** is value mark count for exploding sorts. If no selvar, reads from last list with no variable assigned.

`READT var THEN/ELSE statements`
Reads the next block from tape into ***var***.

`READV{U} var FROM {fvar,} item,a LOCKED stmts THEN/ELSE stmts`
Same as READ, but reads only ***a*** from item. ***U*** locks item.

`READX{U} var FROM {fvar,} item LOCKED stmts THEN/ELSE stmts`
Same as READ, for binary data. ***U*** locks item.

`RELEASE {{fvar,} exp}`
Unlocks the item specified by ***exp***. Without ***fvar*** or exp, unlocks all items locked by this program.

`REM(num,denom)`
Same as MOD.

`REPLACE(da,a{,v{,s}};exp)`
Replaces the specified part of ***da*** with ***exp***.

`RETURN`
Returns from subroutine.

`RETURN TO label`
Returns from internal subroutine to ***label***.

PICK/BASIC STATEMENTS AND FUNCTIONS, continued

`REWIND THEN/ELSE statements`
Rewinds tape.

`RND(exp)`
Generates a random number between 0 and ***exp***.

`ROOT(file,a-id TO var THEN/ELSE statements`
Puts the FID of the root of the secondary index specified by attribute defining item ***a-id*** of ***file*** into ***var***.

`RQM`
Releases time slice. RQM is also a synonym for sleep.

S

`SELECT {fvar} {TO var}`
Selects item-IDs in ***fvar*** into a select-list stored in ***var***, or in the default select-list.

`SELECT da {TO var}`
Selects values from ***da*** into a select-list stored in ***var*** or in the default select-list.

`SENTENCE({n})`
Returns TCL sentence which started program. ***n*** returns only element n of the sentence.

`SEQ(exp)`
Returns the ASCII value of the first character in ***exp***.

`SIN(exp)`
Sine.

`SLEEP{(exp)}`
Suspends the program. ***exp*** can be either a number of seconds or a military time, in quotes. Default is one second.

`SORT(da)`
Sorts ***da*** in ascending order, changing all system delimiters to attribute marks.

`SOUNDEX(str)`
Converts ***str*** into a phonetic code string.

`SPACE(len)`
Generates a string of ***len*** spaces.

`SQRT(exp)`
Square root.

`STOP{en {,p1,p2...}}`
Logical end of program. Displays error message number ***en*** using parameters ***p1***, etc.

`STR(exp,num)`
Generates a string by repeating ***exp num*** times.

`SUBROUTINE {name} {(args)}`
Identifies the program as subroutine ***name***. ***args*** are used to accept information passed by CALL.

`SUM(da)`
Sums numbers at the lowest level of ***da*** and stores them as the next highest level.

`SYSTEM(exp)`
Gets the predefined system value corresponding to ***exp***.

PICK/BASIC STATEMENTS AND FUNCTIONS, continued

T

`TAN(exp)`
Tangent.

`TIME()`
Internal time of day.

`TIMEDATE()`
Time and date in external format.

`TRIM(exp)`
Removes extra spaces from ***exp***.

`TRIMB(exp)`
Removes trailing spaces from ***exp***.

`TRIMF(exp)`
Removes leading spaces from ***exp***.

U

`UNLOCK{exp}`
Unlocks execution lock ***exp*** or all locks set by this program.

W

`WEOF THEN/ELSE statements`
Writes an end-of-file mark to tape.

`WRITE{U} exp ON {fvar,} item`
Replaces ***item*** with contents of ***exp***. U leaves the item locked.

`WRITET exp THEN/ELSE statements`
Writes the value of ***exp*** to tape.

`WRITEV{U} exp ON {fvar,} item,a`
Writes ***exp*** to indicated ***a*** of ***item***. U leaves the item locked.

X

`XTD(exp)`
Converts ***exp*** from hexadecimal to decimal.

PICK/BASIC DEBUGGER COMMANDS

You can start the debugger in three ways:

- Press BREAK.
- Use the RUN verb's (D option.
- Include a DEBUG statement in the program.

These commands are available in the debugger:

`Bvoc{&voc}`	Sets a breakpoint based on the value of variable ***v***. ***o*** is a logical operator and ***c*** is the value to compare v to.
`B$on`	Sets a breakpoint based on line number ***n***. ***o*** is a logical operator.
`D`	Displays the breakpoint and trace tables.
`DE`	Enters the system debugger.
`E`	Toggles instruction stepping.
`END`	Ends the program and returns to TCL.
`G`	Proceeds from breakpoint,
`Gn`	Goes to line ***n***.
`K{n}`	Deletes breakpoint ***n***, or all breakpoints if you omit n.
`L`	Displays source for the current line.
`L{m-}n`	Displays source for line ***n*** or lines ***m-n***.
`L*`	Displays source for the whole program.
`LP`	Toggles output device.
`LT`	Toggles line tracing.
`Nx`	Continues through ***x*** breakpoints before stopping.
`OFF`	Logs off.
`P`	Inhibits program output.
`PC`	Same as PRINTER CLOSE.
`R`	Pops the return stack.
`S`	Displays routine stack.
`T`	Toggles the breakpoint trace table.
`Tv`	Sets variable ***v*** in trace table.
`U{v}`	Removes variable ***v*** from trace table; removes all variables if no v.
`V`	Verifies object code.
`Z`	Specifies source code file and item.
`$ or ?`	Displays the program name, line number and code verification status.
`/m`	Displays array ***m***.
`/m(x)`	Displays element ***x*** of array ***m***.
`/m(x,y)`	Displays element ***x,y*** of array ***m***.
`/v`	Displays the value of variable ***v***.
`/*`	Displays the symbol table.

NOTES

NOTES